
Weekly Report / MEI Honghui

6.19-6.25

ChinaVis Review (main points)

- Some incorrect grammars, typos and minor modification.
- More description around expressiveness, accessibility, efficiency.
- More about “degree of abstraction”
- Discuss data structures related to graphs.

Paper Read

visual programming

[1] M. Takatsuka and M. Gahegan, “GeoVISTA Studio: A codeless visual programming environment for geoscientific data analysis and visualization,” *Comput. Geosci.*, vol. 28, no. 10, pp. 1131–1144, 2002.

[2] T. Curk et al., “Microarray data mining with visual programming,” *Bioinformatics*, vol. 21, no. 3, pp. 396–398, 2005.

[3] C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier, and C. Godin, “OpenAlea: a visual programming and component-based software platform for plant modeling,” *Funct. Plant Biol.*, vol. 35, pp. 751–760, 2008.

[4] B. Sassi and M. Djamel, “Visual Programming and Program Visualization – Towards an Ideal Visual Software Engineering System –,” *Int. J. Inf. Technol.*, vol. 1, no. 3, pp. 56–62, 2011.

[5] T. R. G. Green and M. Petre, “Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework,” *J. Vis. Lang. Comput.*, vol. 7, no. 2, pp. 131–174, 1996.

others

[6] T. Beltramelli, “pix2code: Generating Code from a Graphical User Interface Screenshot,” *no. Nips*, pp. 1–8, 2017.

CNN networks working to generate codes for GUI from simple images.

Australia Visa

Visa has been accepted and granted.

Flights booked.

VisComposer: Revision

Review points:

- Not enough novelty at the point of programmability
 - The idea of shaders is too crude and metaphor to CG pipeline is weak
 - The shaders do not exactly improve usability
 - No concrete evidence/design to proof its ease of use
 - ◆ No more than textual programming when working complex layout (i.e. treemap)
 - Model is unclear
- Not enough evaluation
 - Compare with other tools
 - Need qualitative and quantitative evaluation of usability and efficiency
 - No real-world cases
- Others
 - Need to be published online
 - Lack of capability of interaction and animation

Ideas of Refinement

- A top-down layering and progressive model
 - Scenegraph → workflow → module → code
 - Each level has a complete functionality
 - ◆ Scenegraph: charting software (add functionality such as drag data direct to a scenegraph node to generate a simple form)
 - ◆ Workflow: dataflow system (drag-n-drop binding)
 - ◆ Module: declarative (interactive specification)
 - ◆ Code: code (enhanced coding interface. see <http://esprima.org/>)
- Mechanism to enhance programmability

-
- Code everywhere
 - ◆ Decompose to sub-block (i.e. workflow → modules + code blocks)
 - ◆ Modify-on-demand
 - ◆ Area-of-effect-on-demand (restrict decomposition to a small part and remain origin appearance as much as possible)
 - Highlight the relationships among: data – presentation – specification – codes
 - ◆ HL on hover
 - ◆ Structured (such as a hierarchical list) display
 - ◆ Audit the dataflow and visual mapping process
 - Paper writing:
 - Evaluation
 - Describe a novel model
 - Focus on ease-of-use interactions

Timeline of VisComposer and VisEvo

To be determined after sync with teammates next Mon.

TODO Next Week

- Sync time points with teammates
- VisComposer
 - determine detailed refinement on system interface
 - hand over the code
 - finish writing paper outline